GUIDE FOR BENCHMARK FOR 3D INTEREST POINTS DETECTION ALGORITHMS

There are two steps for evaluating an "interest point detection algoritm" with respect to the interest points marked by human subjects:

1) Building the ground truth
2) Evaluating with respect to the built ground truth

The MATLAB codes implementing the first step are available in the `IP_BENCHMARK\CODES\GROUND_TRUTH` folder, while the codes for the second step are in `IP_BENCHMARK\CODES\EVALUATION`

We provide all the output files in the `IP_BENCHMARK\OUTPUT_DATA` folder. They can be regenerated using the provided code.

We performed the analysis on two datasets, namely Dataset A and Dataset B.

Dataset A consists of 24 models which were hand-marked by 23 human subjects.

Dataset B is larger with 43 models, and it contains all the models in Dataset B. The number of human subjects who marked all the models in this larger set is16.

3D Models

All the models are available in the folder `IP_BENCHMARK\MODEL_DATASET` in MAT files.

Each file contains two matrices:

`V` is a matrix where each row contains the (x, y, z) coordinates of a vertex of the triangular mesh model.

`F` is the face matrix; each row defines a triangle with vertex indices of V.

Human Subjects' Data

The interest points marked by human subjects' are stored in the folder `\IP_BENCHMARK\HUMAN_SUBJECTs_INTEREST_POINTS` as text files.

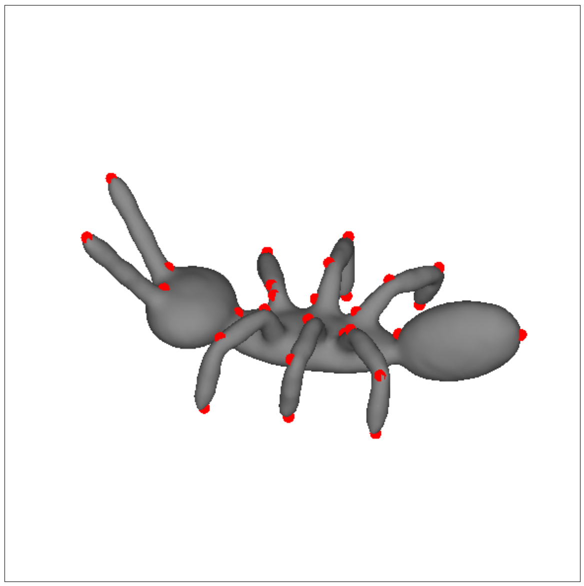
The text files are named with respect to the following template:

`subjectname-modelname_points.txt`

An axample is `jck-ant_points.txt` . Where `jck` is the ailas for the human subject and `camel` is the name of the 3D model. Each row in the text file gives x, y, z coordinates of an interest point marked by the subjet on the 3D model.

The following MATLAB code loads the model `ant` and the interest points marked by subject `jck`, then visualizes `jck`'s points on the model ant.

```
load('C:\IP_BENCHMARK\MODEL_DATASET\ant.mat')
IP_subject =load('C:\IP_BENCHMARK\HUMAN_SUBJECTs_INTEREST_POINTS\jck-ant_points.txt');

figure(1);
trisurf(F,V(:,1),V(:,2),V(:,3),ones(length(F),1),'edgealpha',0);
material dull; lightangle(0,180); lighting gouraud; colormap(gray);
daspect([1 1 1]); axis off; view(0,180); hold on;
plot3(IP_subject(:,1),IP_subject(:,2),IP_subject(:,3),'r.','MarkerSize',30);
hold off;
```



Building the ground truth

The MATLAB codes for building the ground truth are available in the folder `IP_BENCHMARK\CODES\GROUND_TRUTH`.

The main scripts to build the ground truth are `SCRIPT_ground_truth_datasetA.m` and `SCRIPT_ground_truth_datasetB.m`, which merge the interest points marked by human subjects into a single ground truth point set for Dataset A and Dataset B, respectively.

Some lines of the `SCRIPT_ground_truth_datasetA.m` (or `SCRIPT_ground_truth_datasetB.m`) should be modified in order to point to the directories. Check the following before running the code:

1) Our MATLAB code uses, Gabriel Peyre's toolbox for the calculation of geodesic distances on the models. The directories containing the toolbox files should be indicated in the `SCRIPT_ground_truth_datasetA.m` (or

`SCRIPT_ground_truth_datasetB.m`).

```
path(path,'C:\toolbox_peyre\toolbox_graph\');
path(path,'C:\toolbox_peyre\toolbox_fast_marching\');
path(path,'C:\toolbox_peyre\toolbox_fast_marching\toolbox\');
```

Gabriel Peyre's toolbox can be downloaded from
http://www.ceremade.dauphine.fr/~peyre/teaching/manifold/tp3.html

2) Two input directories should be specified pointing to the 3D models and Human Subjects' Interest Points in the `SCRIPT_ground_truth_datasetA.m` (or `SCRIPT_ground_truth_datasetB.m`):

```
% Locate input directories where data will be read from
MODEL_DIR='C:\IP_BENCHMARK\MODEL_DATASET\';
SUBJECT_DATA_DIR = 'C:\IP_BENCHMARK\HUMAN_SUBJECTs_INTEREST_POINTS\';
```

3) Two output directories should be specified. One is `SUBJECT_IP_PARTITIONS_DIR` where intermediate mat files will be stored. Each file contains a model partitioned into geodesic neighborhoods around a subject's interest points. The other is `GROUND_TRUTH_DIR` where final ground truth data is written.

```
% Locate output directories where data will be written to
SUBJECT_IP_PARTITIONS_DIR = 'C:\IP_BENCHMARK\OUTPUT_DATA\SUBJECT_IP_PARTITIONS_A\';
GROUND_TRUTH_DIR = 'C:\IP_BENCHMARK\OUTPUT_DATA\GROUND_TRUTH_A\';
```

4) Set the values of σ , for which ground truth will be calculated (Refer to [1] or the explanation below for the role of σ). The default is as follows:

```
radius_tolerance_factor = [0.01:0.01:0.1];
```

5) Make sure that the MAT files `exp_model_list_A, exp_model_list_B`, `subject_list_A`, and `subject_list_B` are present in the same directory as the script. These MAT files contains cells listing the names of the 3D models and the alias names of the human subjects corresponding to each dataset.

The ground truth of a 3D model is stored in a variable `GT_MODEL` and written to the output folder `GROUND_TRUTH_DIR`.

We have two parameters while constructing the ground truth: We look a consensus of at least n subjects within a region of radius σd_M , where d_M is the diameter of the 3D model.

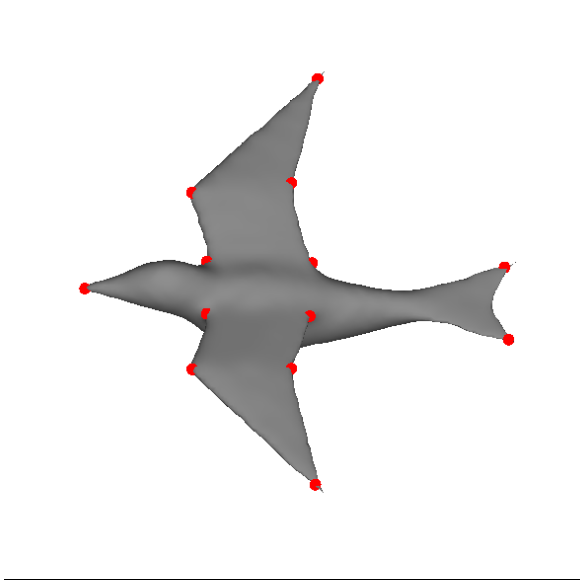
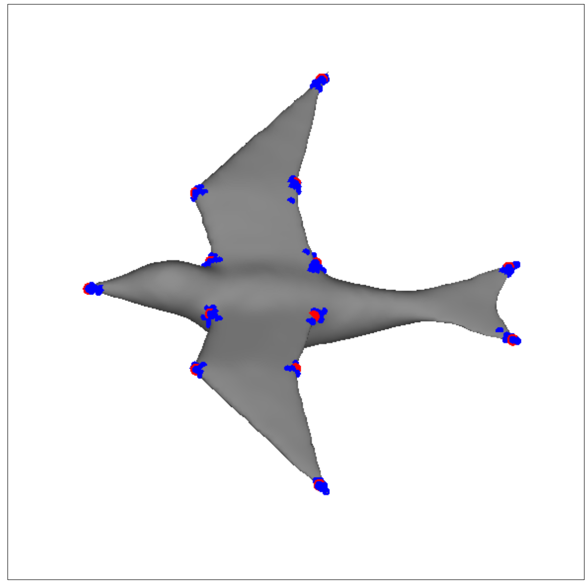
GT_MODEL is a cell of size *T*-by-*(N-1)*-by-*2* , where *T* is the number of different settings of σ, i. e. the length of the vector radius_tolerance_factor , and *N* is the number of human subjects. There is a different set of ground truth points for each setting of the pair (σ_i, *n*), where $i = 1,...,T$, and $n = 2,..,N$. The cell element of GT_MODEL at index (*i, n-1,*1*)* is the list of the vertex indices that constitute the ground truth for the pair (σ_i, *n)*. This list is of length *K*, where *K* is the number of ground truth points corresponding to the setting (σ_i, *n*).

The cell element of GT_MODEL at index (*i, n-1,*2) is another cell of length *K*. If we call this cell as *G*, then the element at index *k* is a string indicating the indices of the vertices that are marked by human subjects in the σd_M vicinity of the kth ground truth point. I.e., these vertices correspond to a clustered group of human marked interest points and the ground truth point is the representative of that cluster (see [1]).

The following MATLAB code reads the GT_MODEL corresponding to the bird_3 model from the folder GROUND_TRUTH_DIR, and extracts and visualizes the ground truth interest points for $\sigma_3 = 0.03$, and $n = 10$ with red dots in Figure 1. The points marked by the human subjects in the vicinity of each ground truth point are shown in Figure 2 as smaller blued dots.

```
GROUND_TRUTH_DIR = 'C:\IP_BENCHMARK\OUTPUT_DATA\GROUND_TRUTH_A\';
load([GROUND_TRUTH_DIR 'bird_3']);
sigma = 0.03; i = 3;
n = 10;
ground_truth_points = GT_MODEL{i,n-1,1} ;
clusters = GT_MODEL{i,n-1,2} ;

figure(1);
load('C:\IP_BENCHMARK\MODEL_DATASET\bird_3.mat')
GT_IP = V(ground_truth_points,:);
trisurf(F,V(:,1),V(:,2),V(:,3),ones(length(F),1),'edgealpha',0);
material dull; lightangle(0,180); lighting gouraud; colormap(gray);
daspect([1 1 1]); axis off; view(0,180); hold on;
plot3(GT_IP(:,1),GT_IP(:,2),GT_IP(:,3),'r.','MarkerSize',30);
hold off;

figure(2);
trisurf(F,V(:,1),V(:,2),V(:,3),ones(length(F),1),'edgealpha',0);
material dull; lightangle(0,180); lighting gouraud; colormap(gray);
daspect([1 1 1]); axis off; view(0,180); hold on;
plot3(GT_IP(:,1),GT_IP(:,2),GT_IP(:,3),'r.','MarkerSize',30);
for c = 1:length(clusters);
    C_c = clusters{c};
    C_I = str2num(C_c);
    S_IP = V(C_I,:);
    plot3(S_IP(:,1),S_IP(:,2),S_IP(:,3),'b.','MarkerSize',20);
end;
```

```
hold off;
```

Evaluating an Interest Point Detection Algorithm

We have evaluated six interest points detection algorihtms:

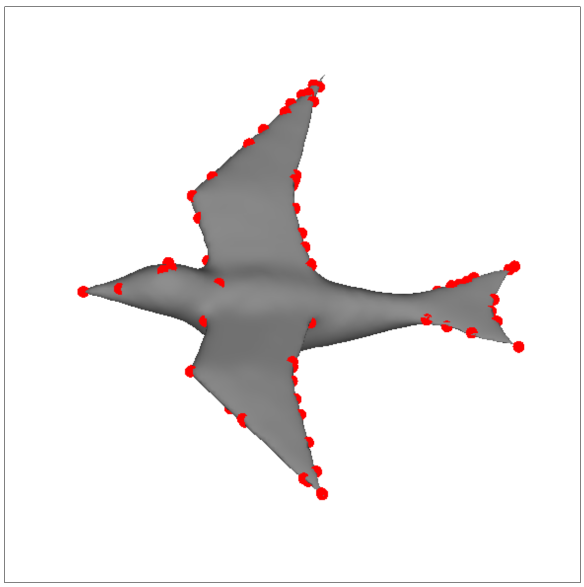
1) Mesh saliency [Lee et al. 2005]
2) Salient points [Castellani et al. 2008]
3) 3D-Harris [Sipiran and Bustos, 2010]
4) 3D-SIFT [Godil and Wagan, 2011]
5) Scale-dependent corners [Novatnack and Nishino, 2007]
6) HKS-based interest points [Sun et al. 2009]

The interest points generated by these algorithms are available in the folders:

```
\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\Mesh_saliency
\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\Salient_points
\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\3D-Harris
\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\3D-SIFT
\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\SD-corners
\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\HKS
```

In each subfolder corresponding to an interest point detection algorihtm, there are MAT files in the names of the 3D models. A MAT file stores a variable `IP_vertex_indices` , a vector listing the vertex indices of the interest points detected by the algorithm. The following code reads and plots the interest points of model `bird_3` detected by Mesh Saliency method:

```
method_name = '3D-Harris';

ALGORITHM_IPs_MAIN_FOLDER = 'C:\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\';
ALGORITHM_IPs_DIR = [ALGORITHM_IPs_MAIN_FOLDER method_name '\'];

load([ALGORITHM_IPs_DIR 'bird_3']);

figure(1);
load('C:\IP_BENCHMARK\MODEL_DATASET\bird_3.mat')
A_IP = V(IP_vertex_indices,:);
trisurf(F,V(:,1),V(:,2),V(:,3),ones(length(F),1),'edgealpha',0);
material dull; lightangle(0,180); lighting gouraud; colormap(gray);
daspect([1 1 1]); axis off; view(0,180); hold on;
plot3(A_IP(:,1),A_IP(:,2),A_IP(:,3),'r.','MarkerSize',30);
hold off;
```



The MATLAB codes for evaluating the Interest Point Detection Algorithm are available in the folder `IP_BENCHMARK\CODES\EVALUATION`.

The main scripts to calculate false positives, false negatives and weighted miss error are `SCRIPT_get_eval_model_datasetA.m` and `SCRIPT_get_eval_model_datasetB.m`. These codes read interest points detected by an algorithm and the ground truth for each model, for Dataset A and Dataset B, respectively.

Some lines of the `SCRIPT_get_eval_model_datasetA.m` (or `SCRIPT_get_eval_model_datasetB.m`) should be modified in order to point to the directories. Check the following before running the code:

1) Determine the method name and comment out the line. The options are as follows:

```
 method_name = 'Mesh_saliency';
% method_name = 'Salient_points';
% method_name = '3D-Harris';
% method_name = '3D-SIFT';
% method_name = 'SD-corners';
% method_name = 'HKS';
```

2) The evaluation code uses Gabriel Peyre's toolbox for the calculation of geodesic distances on the models. The directories containing the toolbox files should be indicated in the `SCRIPT_get_eval_model_datasetA.m` (or `SCRIPT_get_eval_model_datasetB.m`).

```
path(path,'C:\toolbox_peyre\toolbox_graph\');
path(path,'C:\toolbox_peyre\toolbox_fast_marching\');
path(path,'C:\toolbox_peyre\toolbox_fast_marching\toolbox\');
```

   Gabriel Peyre's toolbox can be downloaded from
   http://www.ceremade.dauphine.fr/~peyre/teaching/manifold/tp3.html

3) Three input directories should be specified pointing to the 3D models, Ground Truth files, and Algorithm's Interest Points in the `SCRIPT_get_eval_model_datasetA.m` (or `SCRIPT_get_eval_model_datasetB.m`) :

```
% Locate input directories
MODEL_DIR='C:\IP_BENCHMARK\MODEL_DATASET\';
GROUND_TRUTH_DIR = 'C:\IP_BENCHMARK\OUTPUT_DATA\GROUND_TRUTH_A\';
ALGORITHM_IPs_MAIN_FOLDER = 'C:\IP_BENCHMARK\ALGORITHMs_INTEREST_POINTS\';
```

4) The output directory should be specified where the errors will be written to:

```
% Locate output directory
FN_FP_MAIN_FOLDER = 'C:\IP_BENCHMARK\OUTPUT_DATA\FN_FP_DATA_A\';
```

5) Determine the range for localization error tolerance r :

```
global error_range
error_range = [0:0.005:0.12];
```

6) Make sure that the MAT files `exp_model_list_A`, and `exp_model_list_B` are present in the same directory as the script.

The error rates of an algorithm with respect to a 3D model are stored in a variable `EVAL_MODEL` and written to the output folder `FN_FP_MAIN_FOLDER\[method_name]`.

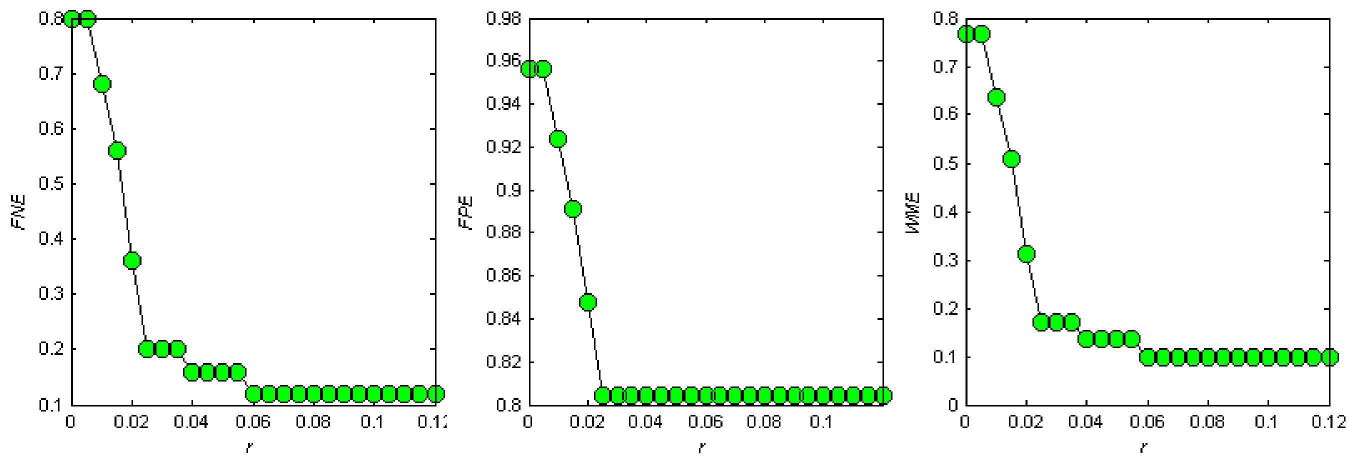
The size of `EVAL_MODEL` is in accordance with the `GT_MODEL`, where ground truth points for different settings of σ and n are stored. `EVAL_MODEL` is a cell of size T-by-$(N-1)$-by-5 , where T is the number of different settings of σ, i. e. the length of the vector `radius_tolerance_factor` , and

N is the number of human subjects. The cell elements at index (*i, n-1,* :) correspond to the following measures with respect to the ground truth points generated with the parameter setting (σ_i, *n*) and stored in `GT_MODEL`(*i, n-1,* 1) :

- The cell element of `EVAL_MODEL` at index (*i, n-1,* 1) : A vector giving the number of false negatives for the error range (range of the localization error tolerance).
- The cell element of `EVAL_MODEL` at index (*i, n-1,* 2) : A vector giving the number of false positives for the error range (range of the localization error tolerance).
- The cell element of `EVAL_MODEL` at index (*i, n-1,* 3) : Number of ground truth interest points for the parameter setting (σ_i, *n*) .
- The cell element of `EVAL_MODEL` at index (*i, n-1, 4*) : Number of interest points detected by the algorithm.
- The cell element of `EVAL_MODEL` at index (*i, n-1, 5*) : A vector giving the Weighted Miss Error for the error range (range of the localization error tolerance).

The following code reads `EVAL_MODEL` for the model `ant` obtained with the Mesh Saliency method. It plots FNE, FPE, and WME calculated with respect to the ground truth interest points for $\sigma_3 = 0.03$, and $n = 10$.

```
FN_FP_MAIN_FOLDER = 'C:\IP_BENCHMARK\OUTPUT_DATA\FN_FP_DATA_A\';
method_name = 'Mesh_saliency';
FN_FP_DIR = [FN_FP_MAIN_FOLDER  method_name '\'];
load([FN_FP_DIR 'ant']);
sigma = 0.03; i = 3;
n = 10;
error_range = [0:0.005:0.12];

E_MODEL = EVAL_MODEL(i,n,:);
E_MODEL = E_MODEL(:);

num_GT_points = E_MODEL{3};
num_alg_points = E_MODEL{4};

FN = E_MODEL{1}/num_GT_points;
FP = E_MODEL{2}/num_alg_points;
WME = E_MODEL{5};

figure(1);
plot(error_range,FN,'-ko','LineWidth',1,'MarkerFaceColor','g','MarkerSize',8);
xlabel('{\itr} ' );
ylabel('{\itFNE}');

figure(2);
plot(error_range,FP,'-ko','LineWidth',1,'MarkerFaceColor','g','MarkerSize',8);
xlabel('{\itr} ' );
ylabel('{\itFPE}');

figure(3);
plot(error_range,WME,'-ko','LineWidth',1,'MarkerFaceColor','g','MarkerSize',8);
xlabel('{\itr} ' );
ylabel('{\itWME}');
```



Notice that the code above extracts and plots FNE, FPE, and WME for a single model. In order to calculate these statistics averaged over the Dataset A or Dataset B, the MATLAB files `SCRIPT_get_FN_FP_statistics_datasetA.m` and `SCRIPT_get_FN_FP_statistics_datasetA.m` can be utilized.

Check the following before running the code:

1) Determine the method name and comment out the line. The options are as follows:

```
 method_name = 'Mesh_saliency';
% method_name = 'Salient_points';
% method_name = '3D-Harris';
% method_name = '3D-SIFT';
% method_name = 'SD-corners';
% method_name = 'HKS';
```

2) Loacte the directory where `EVAL_MODEL` files had been written to:

```
% Locate directory of error rates of models
FN_FP_MAIN_FOLDER = 'C:\IP_BENCHMARK\OUTPUT_DATA\FN_FP_DATA_A\';
```

3) Specify the setting for σ and n :

```
n = 11;
sigma = 0.05;
```

4) Specify the range for localization error tolerance r :

```
global error_range
error_range = [0:0.005:0.12];
```

5) Make sure that the MAT files `exp_model_list_A`, and `exp_model_list_B` are present in the same directory as the script.

References

- Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. In: ACM SIGGRAPH 2005, pp. 659–666 (2005)
- Castellani, U., Cristani, M., Fantoni, S., Murino, V.: Sparse points matching by combining 3D mesh saliency with statistical descriptors. Comput. Graph. Forum 27(2), 643–652 (2008)
- Sipiran, I., Bustos, B.: A robust 3D interest points detector based on Harris operator. In: Eurographics 2010 Workshop on 3D Object Retrieval (3DOR'10), pp. 7–14 (2010)
- Godil, A., Wagan, A.I.: Salient local 3D features for 3D shape retrieval. In: 3D Image Processing (3DIP) and Applications II, SPIE (2011)
- Novatnack, J., Nishino, K.: Scale-dependent 3D geometric features. In: ICCV, pp. 1–8, (2007)
- Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Eurographics Symposium on Geometry Processing (SGP), pp. 1383–1392 (2009)